Tilburg University

An extensible business communication language

Weigand, H.; Hasselbring, W.H.F.

Published in:
International Journal of Cooperative Information Systems, IJCIS

Publication date:
2001

Link to publication in Tilburg University Research Portal

Citation for published version (APA):
Weigand, H., & Hasselbring, W. H. F. (2001). An extensible business communication language. *International Journal of Cooperative Information Systems, IJCIS*, *10*(4), 423-442.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 05. May. 2023

An Extensible Business Communication Language

Hans Weigand & Wilhelm Hasselbring

Infolab, Dept. Information Management and Computer Science, Tilburg University,
PO Box 90153, NL-5000 LE Tilburg, Netherlands,
Email: {weigand|hasselbring}@kub.nl

Abstract. A main problem for electronic commerce, particularly for business-to-business applications, lies in the need for the involved information systems to *meaningfully* exchange information. Domain-specific standards may be used to define the semantics of common terms. However, in practice it is not easy to find those domain-specific standards that are detailed and stable enough to allow for real interoperability. Therefore, we propose an architecture that allows for incremental construction of a shared repository including a multilingual thesaurus, which is used in a business communication language. Communicating information systems then refer to the common thesaurus while exchanging messages. Our emphasis is be on separating semantics (in the thesaurus) and syntax (in XML). Therefore, our extensibility is not only that of XML, but also the extensibility of the semantics that is modeled in the shared repository.
The business communication language XLBC is presented and how it can be used in electronic commerce applications. XLBC message patterns and conversation protocols are stored in the shared repository as well.

1 Introduction

In spite of all the initial hype and subsequent disappointment, it is becoming increasingly clear that electronic commerce is taking off on a global scale, not only in the consumer market but also in business-to-business and business-to-administration application areas. However, there are also many barriers that still need to be taken. One barrier is the standardization of the message formats for business communication. Although business-to-business electronic commerce has a longer tradition of electronic data interchange in the form of EDIFACT, it is generally recognized that EDI is too costly and not flexible enough to cope with the dynamics of the new economy [KM97,KL96,MG98]. However, traditional EDI is often being re-examined to define the *meaning* of the transferred data (semantics), and XML is employed as the practical foundation in which to structure this information (syntax). XML is a markup language for creating self-descriptive data; in contrast to HTML, it separates style and content and is extensible in the sense that new tags can be used as long as they are defined in the DTD (document type definition). For electronic commerce, it is especially interesting that one format can be used both for electronic messages (to be processed by computers) and for human interfaces; an XML document itself is already, to some extent, readable for humans (what an EDI document is not), but especially when it is accompanied by a style document (XSL), it can be presented by means of a web browser in any desired layout. This feature not only allows to have one single interface to application systems (for humans and for systems), but also enables hybrid set-ups in which humans and systems are involved in different stages of the process and the same format can be used throughout.

However, XML on itself will not do the job. The receiving party can recognize something as a valid XML document, and when it has the accompanying DTD, it can check whether it adheres to this DTD, but nothing is said yet about the meaning of the data elements. If every company were to develop its own DTDs, there would be no real interoperability. So, although XML is technically superior to EDI X.12, it does not solve the huge problem that EDIFACT has worked on for years, namely, how to define the contents of the messages. What elements should be there, how are they represented and what do they mean? If XML should be used in business-to-business electronic commerce, something equivalent to the EDIFACT standards must be in place. The standardization of messages can be at different levels: at the lexical level of character sets (data representation), the syntactical level of message structures, to a deeper semantic level of vocabulary and integrity constraints. If communicating parties want true communication, they must agree not only on the form but also on the meaning of the messages. It is not necessary and even undesirable to strive for explicit agreement on all semantics. All that is needed is that confusion is avoided and that messages can be processed automatically at both ends - which means that a mapping can be made to the local schema. We call this a minimalist approach.

From an institutional point of view, standards are vehicles for facilitating coordination of economic activities [H$^+$95]. Instead of repeated coordination between actors, a standard solves a number of dilemmas for actors in a situation where communication is required. A standard therefore diminishes the need for ad-hoc coordination. On the other hand, there is an increased need for concerted action when standards are created or changed. Normally, this concerted action is performed at the level of standardization committees. However, at present this often turns out to be infeasible, or only feasible to a very limited extent. In today's open and dynamic business environment, the partners have to take over part of the standardization process to themselves. This can involve two or more partners who intend to set up a business relationship on the spot, or an industrial platform/market owner who does this standardization for its members. For such a setting, a flexible system architecture that allows for dynamic evolution of the business communication language is required.

In this paper, we propose an architecture that allows for incremental construction of a shared repository, which is used in a business communication language. Our emphasis is on separating semantics (in the repository) and syntax (in XML). In this way, the architecture not only supports extensibility of message syntax (as offered by XML), but also the extensibility of the message semantics (by means of the shared repository).

2 Foundations for Formal Business Communication

Communication languages have been defined in several environments: KQML in the AI community, ACL in the FIPA agent consortium, and FLBC in the business communication field. In this section we give a short overview of FLBC that we take as our starting point. We also indicate at what points we have deviated from FLBC while developing XLBC.

2.1 FLBC

A number of researchers have investigated the possibility of developing general-purpose formal languages for business communication (FLBC), notably Kimbrough, Moore, Covington and Lee. The impetus for this research has been a common assessment of the fact that existing EDI standards leave much to be desired in flexibility, in expressivity, in clarity, etc. Kimbrough & Moore mention two assumptions of the FLBC approach [KM97]. The first assumption states that a properly designed FLBC should permit business messaging to begin and to proceed without the business partners having to come to a separate and specific agreement concerning the content, structure, and proper interpretation of the messages to be exchanged. This assumption is very close to the approach called Open-EDI [KL96]. It does not require that every message be based entirely on public lexicons. Exchange of particular vocabularies should certainly be allowed, as should 'linguistic bootstrapping' (agreement to define new expressions in terms of existing expressions). The second assumption calls for a semantic foundation of the language in First-Order Logic.

FLBC is based on speech act theory that makes a distinction between the illocutionary force of a message and the propositional content [KM97,Moo99]. By explicating the illocutionary force, FLBC makes clear that messages are not just pieces of data, but (intend to) have some social effects, such as creating an obligation. Moreover, the propositional content is represented in such a way that it contains indeed a proposition, that is, a statement that can be logically true or not (in the case of a assertive message), or an action to be taken (in the case of a directive message). This is in contrast to traditional EDIFACT messages where all the necessary data elements are present (otherwise it would not work, as a matter of course), but not structured in the form of a proposition or action. As a result, the syntax definitions of traditional EDI are somewhat arbitrary and unpredictable. In the FLBC approach proposed by Kimbrough & Moore, the basic structure of FLBC messages is defined once for all. Of course, different message types (also called patterns) can be defined on this basis, such as for ORDER, INVOICE, etc. These message patterns differ in the actions that they refer to and the arguments that these actions take. However, they can always be parsed, and interpreted to some extent; for the full interpretation, the receiver should know the meaning of the terms and predicates.

A simplified example of an FLBC message is the following order message:

```
msg([person(jbouw)],person(vsues),request,
         deliver(vsues,p2,jbouw),
         product_ean(p2,5012345678900),
         quantity(p2,900)),
[], msg128576)
```

The first two arguments are the Sender (`jbouw`) and Receiver (`vsues`), followed by the illocution (`request`). The propositional content is a set of logical clauses. In contrast to earlier languages such as EDIFACT, formal semantics are considered important in FLBC in order to arrive at rigorous definitions and facilitate automatic processing. FLBC uses First-Order Logic as much as possible. XLBC is based on communication semantics described in [WVD95] called Illocutionary Deontic Logic, which is a modal logic. The following subsections summarize our main extensions and adaptations to FLBC.

2.2 Conversation objects

Business messages that are exchanged typically occur in conversation patterns. For example, an order is followed by an acknowledgment, and together they form a transaction. It is not the order itself, but the order transaction that creates the obligation for the other party to deliver, as expressed in the legal definition of a purchase order as a "written authorization for a supplier to ship products at a specified price which becomes a legally binding contract once the supplier accepts it." In [WvH99], a pattern language is described that distinguishes different levels of conversations. Here, we do not repeat the details of this framework, but just assume that conversation objects at different aggregation levels can be defined. In section 4.5, we will describe one level, the level of tranactions.

2.3 Roles

In line with linguistic theory, Kimbrough takes a predication/argument structure as the basic representation of events [Kim98]. This means that events are thought of as a special kind of entity of a certain type (e.g., delivery) and the arguments are the entities that play a role in the event (who delivers, what is delivered, etc). The role names are taken from a controlled set. In linguistics (e.g., [Dik89]), a set of about 10 to 20 role names are distinguished, but we allow in principle any role name to be added. A standard example is an event of delivery, where three roles can be distinguished: the agent, who does the delivery; a theme, the thing that is delivered; and a destination, at which the thing is delivered. Besides the mandatory roles that can be associated with an event, there are also optional roles, such as Time, Location and Beneficiary. In [Wei89] we have shown that Dik's framework – Functional Grammar – provides a suitable basis for knowledge representation. In line with Functional Grammar [Dik89], we allow a predication to have restrictors in addition to the role arguments. Restrictors further identify the entity or event denoted by the predication. Using restrictors, it is possible to add any attribute to the predication, as long as it is semantically coherent. Predication restrictors are the semantic equivalent of adverbial expressions in natural language.

2.4 Illocutions

The illocutions distinguished in FLBC include Assert, Request, Commit, Ask etc. Illocutions are distinguished from message types, such as Invoice, Delivery-order, Customs-clearing or Quotation. We distinguish illocutions from message types for two reasons. Firstly, because many message types must be analyzed as consisting of more than one speech act. Secondly, because the illocution can be the same, but the propositional content different, and in that case we also want to distinguish the messages type. An invoice differs from a delivery-order in the action to be taken: the action to pay versus the action to deliver, but both have the illocution ‘Request’.

It is important that the illocutions and message types are well-defined logically. This will be done later in the development of the FLBC component library. It should be kept in mind that the list of illocutions is neither closed once for all — there may arise a need for yet another illocution to be distinguished — nor arbitrary — there is a well-thought-out logical structure behind it.

2.5 Predicates

FLBC does not say much about the structure of the lexicon in which predicates are defined. With XLBC, the lexicon – or thesaurus as we call it – is set up in a linguistically motivated way and with support for multilinguality. Predicates can be nominal, adjectival or verbal. Verbal predicates denote some action or activity. Predicates with their roles and possible selection restrictions are called predicate frames and are stored in the multilingual thesaurus. Some are very general (e.g. deliver, arrive), and some will be domain-specific. The thesaurus entries specify lexical information with the representation in one or more languages and semantic information.

3 The Shared Repository

XLBC is based on a separation of syntax (XML messages) and semantics. The semantic definitions are stored in a shared repository. This could be made public, but it could as well be restricted in use for one market only. The shared repository contains two parts: a multilingual thesaurus, and an XLBC component library. The former defines the elements of which messages are composed, and the latter defines the messages and higher aggregation structures. We stress that the shared repository distinguishes several component levels, and is not just a large set of possible DTDs. We strongly believe that such a structure is necessary in order to obtain real extensibility.

3.1 The Multilingual Thesaurus

The different elements that make up a message refer to certain real-world entities, such as the parties involved, the products/goods that are exchanged, etc. Usually, these items will have different attributes such as location and price. Both the attributes themselves and the values of those attributes have a lexical representation, such as `"company"` or `"product"`. These representations are basically words that denote some concepts. The thesaurus makes a distinction between words (or lexicals) and concepts, and supports a semantic network of concepts.

Figure 1 illustrates our general communication architecture, in which information systems communicate by means of a common business language (in our case XLBC messages that contain references to concepts in the multilingual thesaurus). The multilingual thesaurus for this language is managed by the vocabulary server, which allows for dynamic extension of the terms used in the business language. The communicating information systems are informed of changes to the common thesaurus by means of a notification service. The shared repository also manages the XLBC Document Type Definitions (see section 4).

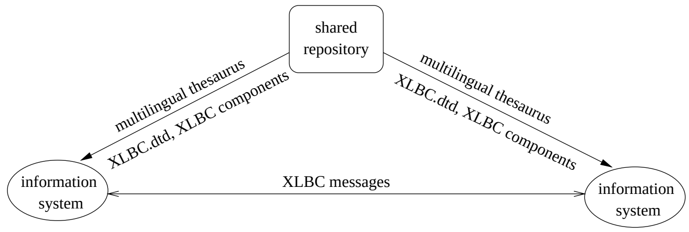

Fig. 1. The general communication architecture. The shared repository manages the common terms that are used in XLBC messages to be exchanged by the communicating information systems.

The thesaurus is built up around a semantic network of concepts. The concepts are defined through their relationships with words and other concepts. Words are the natural language representations of concepts. Multiple words can describe one concept (author and writer may be *synonyms*), and one word can be used to describe multiple concepts (company is a *homonym* that describes both an industrial organization or just a group of friends). We call the relation between a concept and a word a denotation. In addition, concepts are interrelated. For instance, the concept ”author” may have as a generalizing parent (*hyperonym*) a concept that represents a person. This says something about the concept of author. A concept may have different types of relations with other concepts. As in typical object-oriented modeling techniques [Mey97], the parent relation (specialization/generalization), the part-of relation, and non-hierarchical relations between concepts (e.g., in the form of predicate frames) are used for defining concepts in the thesaurus. As can be seen in Figure 2, elements in the different message components that make up the conversation shall link to concepts which have been defined in the semantic network of the thesaurus.

Concepts are part of a semantic network but always organized in generalization hierarchies to facilitate top-down access. We have found it useful to distinguish three levels in this hierarchy. The *top level* contains categories such as Event, Physical Entity, Geographical Entity, Measure Unit, Agent, and Time. It may also contain sub-categories, such as Transport Event, Transform Event, and Transfer Event. The middle level is called the *basic level* and it contains the concepts that are closest to human experience, such as Deliver (a Transport event), Pay (a Transfer event), Day, Month, City, etc. The *bottom level* is made up of sub-concepts, or specialized concepts, such as the many forms of delivery and payment. For retrieving information from the thesaurus, one usually starts at the basic level.

The thesaurus should also provide meaning definitions as far as these are relevant for the business application. Remember that we adhere to a minimalist approach as far

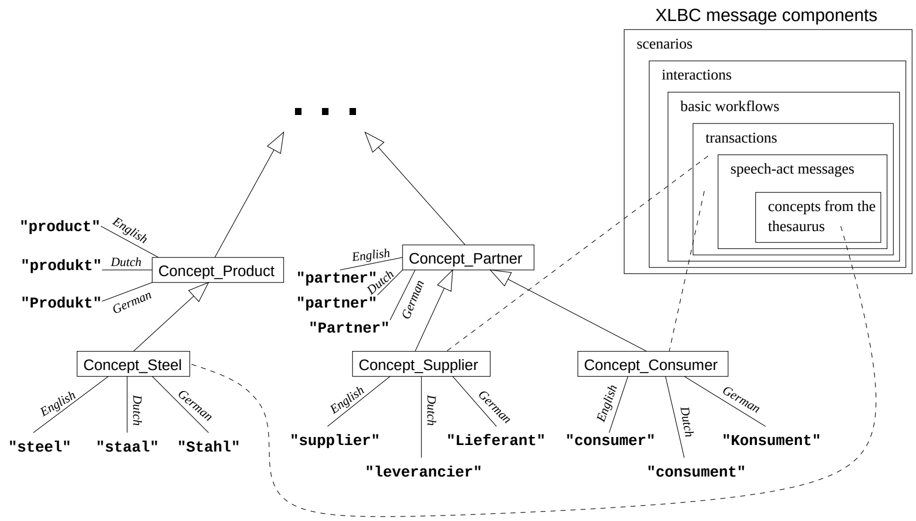

Fig. 2. Linking the XLBC components with the concepts in the semantic network of the multilingual thesaurus. Exemplary, translations of some concepts to English, Dutch and German are given.

as meaning definitions are concerned. We do not have one standard definition method because the relevance can differ and not all concepts are the same. Thus, our approach is more pragmatic than the usual approach in formal ontologies. Usually, real-world concepts (natural kinds) such as House, Horse, or Car, defy a formal definition but for the human interpreter, a verbal meaning definition (as in an ordinary dictionary) is sufficient, whereas the machine does not need to reason about them. For event types, it is often sufficient to specify the category (for example, Pay is a Transfer event) and the selection restrictions on the arguments (in this case, a Money theme). Sometimes, additional preconditions and postconditions are useful. For message types, the operational meaning typically depends heavily on the context. For example, an Invoice can be defined as a claim for payment in return for goods delivered or for a service provided. However, whether the Invoice causes an obligation to pay or presupposes it, is not so clear; in fact, different options can be chosen, leading to different trade scenarios. Trade scenarios are represented in the component library, and it is at this place where the operational semantics of message types such as Invoice are to be found (see the next subsection). In the thesaurus, a verbal definition is sufficient then. Although real-world concepts usually do not need a complete definition, sometimes quality regulations are relevant. For example, a product may only be called ‘chocolate’ if it contains at least a certain percentage of cocoa. Although this is not yet possible in the current implementation of the thesaurus, we intend to allow the inclusion of such regulations, but not as part of the definition, but as what they are: rules on the way concept names are applied to instances. In this way, we not only separate syntax and semantics, but also lexical semantics and lexical pragmatics. For the representation of rules, we need an expressive knowledge representation such as is used in logic-based ontologies (cf. [D^+98]).

3.2 The XLBC Component Library

XLBC is based on speech act theory. Furthermore, the XLBC messages have been grouped into different aggregation levels of conversations, as illustrated in Figure 2. At each composition level, various patterns can be defined. As explained briefly in section 2.2 and in detail in [WvH99], speech acts typically go in pairs. The request/accept transaction is an example of a pattern at transaction level. Transactions can grouped into basic workflows, of which reciprocal interactions can be construed. It is possible to specify rules on, for instance, the sequence order in which elements of XLBC-patterns must occur.

The thesaurus and component library provide an architectural framework that is specifically aimed at extensibility. It is unrealistic to assume that an organization could arrive at an exhaustive repository of standard components and standard messages that any industry can adopt. This should not be the aim. Instead, what the repository should provide is a (well-structured) set of building blocks by means of which parties can develop new components with relative low effort. Given a shared repository with a certain content, users can develop new business processes top-down, by taking an existing business process and adapting it according to their needs, or bottom-up, by taking the basic terms of the Thesaurus - perhaps extended with newly defined ones - and compose them into messages etc. The advantage of the XLBC approach - and FLBC approaches in general - is that it supports compositional semantics. For example, if a new kind of business action has been defined - let's say, ‘review’ instead of ‘deliver’ - then it can be combined without further effort into a request for reviewing, a commit to reviewing, or a report on the reviewing being finished. These message types don't need to be defined but can be understood from the meaning of the illocution and the meaning of the business action. When the Thesaurus is owned and managed by a professional community instead of one party, the procedures for updating and extending the shared repository are equally important. In the conclusions, we will have a few suggestions on this point, but the issue as such is beyond the scope of this paper.

4 The Extensible Language for Business Communication XLBC

Below, we introduce the language XLBC (Extensible Language for Business Communication). XLBC combines the semantic orientation of FLBC with the extensible syntax of XML. The Extensible Markup Language (XML) is *subset* of SGML that is designed to make it easy to interchange structured documents over the Internet [McG98]. The main role of XML (as opposed to HTML) in interoperable systems is likely to be for defining the structure of data to be exchanged between heterogeneous information systems. The syntactic structure of XML documents is specified by a Document Type Definition (DTD), which may be thought of as a schema of the document. Instances of XML documents can only be understood in relationship to their DTD. When we talk about XML documents we need to refer to explicit DTDs.

The top-level DTD for XLBC messages is defined as follows:

```
<!ELEMENT MESSAGE       (SPEECH-ACT+, CONTEXT) >
```

This definition defines a message as consisting of one or more speech acts and a context element. The attributes are omitted for reasons of space.

4.1 Speech Act and Propositional Content

Speech act and propositional content is then defined as follows:

```
<!ELEMENT SPEECH-ACT (CONTENT)>
<!ATTLIST SPEECH-ACT
                TYPE            CDATA                #THESAURUS>

<!ELEMENT CONTENT (PREDICATION | COMPLEX-PRED)>

<!ELEMENT PREDICATION (ARG+, RESTRICTOR*)>
<!ATTLIST PREDICATION
                ID              CDATA                #REQUIRED
                TYPE            CDATA                #THESAURUS
                OPERATOR        ('NEG'|'POS')        #IMPLIED
                ASPECT  ('INTEND'|'START'|'CONTINUE'|
                         'STOP'|'DONE')              #IMPLIED>

<!ELEMENT ARG (TERM | PREDICATION | TIMEREF)>
<!ATTLIST ARG
                ROLE            CDATA                #THESAURUS>
```

These definitions contain the basic structure of the speech acts. A speech act is divided into an illocution and a propositional content. The speaker and addressee of the speech act are already determined by the sender/receiver of the message. The propositional content consists of a number of predications, where a simple predication takes the form of a predicate (usually a verb, such as ‘deliver’) followed by one or more arguments. An argument consists of its identifying role and a term. We also allow for sub-clauses, and therefore instead of a term, it is also possible to fill an argument recursively with a predication. The `ID` of a predication is the identifier of the action occurrence, as it is used in event semantics. The `TYPE` attribute defines the message type taken from the controlled vocabulary in the multilingual thesaurus (specified via the keyword `#THESAURUS`). The predicate operator can be positive or negative (default: positive). The aspect operator can be used to indicate a phase of the event (‘going to v’, ‘start v-ing’, ‘is v-ing’, ‘stop v-ing’, ‘has v-ed’). A complex predication allows for Boolean combinations of simple predications.

4.2 Context

The context of the message contains all kinds of pragmatic features, such as the session of which the message is a part, a link to a previous message, but also the thesaurus used or the preferred language setting. This definition is similar to the ones used in FLBC [Moo99] and FIPA [Fou].

```
<!ELEMENT CONTEXT (#PCDATA) >
<!ATTLIST CONTEXT
                SESSION         CDATA     #IMPLIED
                CHECKPREV-MSG   CDATA     #IMPLIED
                THESAURUS       CDATA     #REQUIRED
                LANGUAGE        CDATA     #THESAURUS>
```

4.3 Terms and References

A term is an expression by means of which the speaker refers to some entity [Dik89]. If the entity has a unique identifier, then the reference is simple, comparable to the reference by means of a personal name in natural language. However, this is not always the case. For example, when a customer orders 3 items of product X (identified by some EAN code), then the reference includes both a product type (identified by the EAN code) and a quantity. The situation becomes more complex when the product is not sold in discrete items. In that case, some unit-of-measure is needed, for example, 200 kg. The entity type can be uniquely identified by an EAN code, but it can also be described by means of a general entity type and a list of restrictors. For example, the entity type can be ‘Toyota Carina Model 1432’ and restrictors can specify the color, the transmission system, etc.

```
<!ELEMENT TERM  (NAME?, RESTRICTOR*>
<!ATTLIST TERM
                TAG             CDATA                 #REQUIRED
                TYPE            CDATA                 #THESAURUS
                CODE            CDATA                 #THESAURUS
                QUANTITY        CDATA                 ‘1’
                UNIT            (ITEM| KG|CM |M | ..)  ‘ITEM’>

<!ELEMENT NAME EMPTY>
<!ATTLIST NAME
                OID             CDATA                 #REQUIRED
                CODE            CDATA                 #THESAURUS>


<!ELEMENT RESTRICTOR (#PCDATA)>
<!ATTLIST RESTRICTOR
                ATTR-NAME       CDATA                 #THESAURUS
                DOMAIN          CDATA                 #THESAURUS
                UNIT            CDATA                 #THESAURUS
                OPERATOR        (=|<|>)               ‘ITEM’>
```

A term is defined as an optional name element followed by zero or many restrictors. In the minimal case, we only have the term attributes: tag, type and code. The most important one is the type: it specifies the concept type of the object referred to, for example, ‘money’ or ‘brick’. The concept types are stored in the multilingual thesaurus (either in the general domain or in a specific domain). The domain from which the type is taken is indicated by means of the ‘CODE’ attribute. It is also possible that the concept type is defined by means of some external standard, such as the EAN product code.

The tag attribute is used to identify the term within the context of the XML document and can be used elsewhere in the document (co-reference).

The quantity and measure-unit attributes take care of the quantitative aspect of the term. Note that the objects referred to can be both discrete (and countable), or non-discrete (mass-terms). If they are conceived as being non-discrete, a measure unit must be given.

Restrictors give further qualifications of the term reference. We have opted for a representation in terms of attribute/value pairs. Attribute names are color, weight, seize, price, etc. The possible values of the attribute are taken from some domain, such as the domain of colors (to be more precise, the ISO definition of color names, or some other formalization), the domain of money, etc. In some cases, the value is numeric, but note that this again implies the use of a measure unit (3.5 meter, 500 pound, etc). We also give (limited) opportunity to provide not just a value, but relationships with some values (e.g., ‘higher than 3.5 meter’).

4.4 An Example XLBC message for Hotel Booking

To illustrate the differences between XLBC and more common (non-semantic) approaches, we give a simplified example. The following XML message is based on a demo DTD provided by BizTalk and has the effect of booking a hotel room.

```
<?xml version="1.0" ?>
<BizTalk xmlns="urn:schemas-biztalk-org:biztalk/BizTalkv1_0.xml">
 <Route>
 <From locationID="http://www.kub.nl/send.asp" locationType="HTTP"
  process="Hotel_Booking" path="" handle="0" />
  <To locationID="http://www.superdeluxe.nl/recv.asp" locationType="HTTP"
  process="Hotel_Booking" path="" handle="2" />
  </Route>
 <Body>
 <DMO_BookRoom_1 xmlns="urn:schemas-biztalk-org:biztalk/dmo_bookroom_1.xml">
 <guestInfo>
  <contact name="Hans Weigand" email="h.weigand@kub.nl"
   phone="013-4662806" />
  <rewardsID memberSince="1998-01-01">90023-345</rewardsID>
  </guestInfo>
 <hotel>
  <hotelName>Hilton</hotelName>
  <phone>020-4502001</phone>
  <address>
  <street>Kerkstraat 13</street>
  <city>Amsterdam</city>
  <country>Netherlands</country>
  <stateOrProvince>Noord-holland </stateOrProvince>
  </address>
 </hotel>
 <reservationInfo>
  <city>Amsterdam</city>
```

```
  <country>Netherlands</country>
  <stateOrProvince>Noord-Holland</stateOrProvince>
  <roomInfo roomStyle="single" smoking="0">
  <beds bedStyle="king">1</beds>
  </roomInfo>
  <plannedStay arrival="2001-7-30" nights="7" />
  </reservationInfo>
 <confirmation>
  <creditCard cardType="visa" expires="2002-06-01">12**/creditCard>
  <confirmDate>2001-5-30</confirmDate>
  </confirmation>
  </DMO_BookRoom_1>
  </Body>
  </BizTalk>
```

The message describes the guest data, the hotel data, reservation data, and credit-card data. Such a message is sufficient in normal business situations. However, from a semantic point of view, it is not clear what the status is of this message (is it a request?), what the required action is (stay in the hotel), and who is going to pay, etc. The equivalent XLBC message is now presented in a abbreviated version.

```
<?xml version="1.0" ?>
 <BizTalk xmlns="urn:schemas-biztalk-org:biztalk/BizTalkv1_0.xml">
 <Route>
  ..
  </Route>
 <Body>
 <Message msg-id="128576" msg-type="BookingRequest" date="20010508">
 <Sender Tag="x1" Type="University">
  <Restrictor Attr-name="Name" Domain="ASCII">KUB</Restrictor>
  ..
  </Sender>
 <Receiver Tag="x2" Type="Company" Code="OBI">
  <Name OID="12647758878" Code="EAN" />
  <Restrictor Attr-name="Name" Domain="ASCII">Superdeluxe BV</Restrictor>
  ..
  </Receiver>
 <Speech-act Illocution="Request">
 <Content>
 <Predication ID="e1" Pred="Stay">
 <Arg Role="Agent">
 <Term Tag="x3" Type="Person">
  <Restrictor Attr-name="Name" Domain="ASCII">Hans Weigand</Restrictor>
  ..
  </Term>
  </Arg>
 <Arg Role="Theme">
 <Term Tag="x4" Type="Room">
  <Restrictor Attr-name="Hotelname" Domain="ASCII">Hilton</Restrictor>
```

```
  <Restrictor Attr-name="NumberOfBeds" Domain="Real">1</Restrictor>
  ..
 <Arg Role="Time">
 <Term Tag="x9" Type="TimeInterval" Unit="Day">
  <Restrictor Attr-name="Start" Domain="Date">20010730</Restrictor>
  <Restrictor Attr-name="End" Domain="Date">20010806</Restrictor>
  </Term>
  </Arg>
  </Predication>
  </Content>
  </Speech-act>
 <Speech-act Illocution="Promise">
 <Content>
 <Predication ID="e2" Pred="Pay">
 <Arg Role="Agent">
  <Term Tag="x1" Type="University" />
  </Arg>
 <Arg Role="Recipient">
  <Term Tag="x2" Type="Company" Code="OBI" />
  </Arg>
 <Arg Role="Theme">
 <Term Tag="x5" Code="OSI" Type="Money" Quantity="650" Unit="EURO">
  <Restrictor Attr-name="Cardtype" Domain="ASCII">Visa</Restrictor>
  ..
  </Term>
  </Arg>
  </Predication>
  </Content>
  </Speech-act>
  <Context Language="ENGLISH" Prev-msg="128575" Session="ABC170" />
  </Message>
  </Body>
  </BizTalk>
```

Firstly, The XLBC message describes the sender and receiver of the message. The tags x1 and x2 can be reused later to indicate the agent and recipient of the payment. The message consists of two speech acts, a request and a promise. The request is a request for a certain person to stay in a room (specified with some desired characteristics) for a certain period. The promise is a promise that the university will pay to the hotel company a certain amount of money by credit card.

The XLBC message is longer than the original BizTalk message. This is because the semantic structure is made explicit. Note that as a consequence, the message is also more generic. In the original message, the creditcard data were assumed to be the creditcard data of the guest. However, this is not necessary. The XLBC message makes explicit who is the payer, and allows it to be the guest or the sender or another agent. The hotel may have certain rules on the payment, and this might include that the guest himself should pay. However, the message format does not exclude other scenarios.

4.5 Transactions

Using XLBC, not only message patterns can be defined and stored in the shared repository, but also conversation protocols. The lowest level is called *t*ransaction. A transaction is defined as a minimal sequence of messages that has some effect in the social world. This effect can be the creation of an obligation, or an authorization, agreement that an action has been performed, etc. Transactions can be composed to build workflows and interactions. The following example (without XML syntax) contains a quotiation transaction.

```
transaction quotation (buyer x, seller y, product p)
elements        quote(y,x,deliver(y,x,p),q), accept-quote(x,y,q)),
                cancel, sub-transaction(proposals(x,y,p))
constraints     AFTER( quote(y,x,deliver(y,x,p),q),
                accept-quote(x,y,q) OR proposals(x,y,p) OR cancel)
goal            accept-quote
exit            cancel
end transaction

sub-transaction proposals(party x, party y, issue z)
elements        propose(i,j,z,P),accept-propose(i,j,z,P),
                cancel(x,z)
constraints     AFTER(propose(i,j,P),accept-propose(j,i,P) OR
                propose(j,i,P2) OR cancel)
goal            accept-propose
exit            cancel
end sub-transaction
```

The elements of the transaction are the speech acts that can be exchanged or subtransactions. No strict ordering is given, but only the temporal constraints that should be obeyed. A transaction can be closed successfully or not succesfully. This depends on the last message exchanged.

The subtransaction concept has been added to allow for discussions inside transactions. They have the same structure as transactions. They do not necessarily have a direct effect on the social world, and can only exist in the context of a (normal) transaction. Note that the subtransaction in this case allows for unbounded exchange of proposals

5 Related Work

In the absence of a complete and comprehensive set of document formats, as EDIFACT intended to provide, several attempts are made to set up repositories of components that can be taken out and used by business partners. XML.ORG, for instance, aims at being an independent industry portal for the standardization of XML applications in electronic commerce, whereby it serves as a reference for XML DTDs. BizTalk.ORG is a competing industry initiative started by Microsoft. Commerce One's CBL defines a set of building blocks. These building blocks are then pulled together to make the actual documents describing the interactions between two organizations. xCBL defines these buildig blocks, such as ”address” or ”price”, in XML, and also provides a basic set of standard messags such as invoice. The main difference with XLBC is that the messages have no compositional semantics (cf. the example biztalk message in section 4.4). In fact, initiatives such as xCBL and cXML (Ariba) do not address the semantic layer at all - they are at the same (syntactic) level as EDIFACT, although the techniques are much more advanced. However, it is possible to combine XLBC and xCBL and use the latter as syntax for standard business terms such as price and address.

OASIS is a non-profit international consortium dedicated to accelerating the adoption of product-independent formats based on public standards, notably XML. Oasis could serve as a host for an XML registry and repository, including XLBC. In the short history of XML, much work has been done to define XML messages at a higher level of abstraction, using standards such as RDF Schema, XML Schema and ontologies such as OIL ([OIL]). There is no doubt that such an abstraction from syntax is very useful. However, the semantics in these approaches is most often based on a traditional object-oriented way of thinking and built up from class definitions possibly (in the case of OIL) extended with constraints and rules in the form of axioms. Whereas the semantics provided by XLBC is based on linguistics and speech act theory which in our view fits better with the goal of modelling business communication.

[Lee98] suggests the use of a central repository in which formal trade procedures can be stored. Users can download these trade procedures — formally represented as Petri-Nets — adapt them if necessary, and then adopt them immediately for execution. The XLBC approach goes further by providing not only trade procedures (corresponding to XLBC components at the workflow level), but also term definitions and message types. [G$^+$99] proposes a central repository of standard contracts that can be used by negotiating partners in the process of contract building. [Hue98] advocates a Trading Partner Agreement in which business partners describe a new business process. The definition can be exchanged by means of EDIFACT meta messages. However, this scenario makes not clear yet how the definitions are managed. Moreover, it requires that the message formats are adapted each time a semantic change is made, such as the addition of a new action. In our approach, the message format can be kept unchanged.

A somewhat older approach that bears similarity to XLBC is the Basic Semantic Repository (BSR) [ISO] and the Business System Interoperation (BSI) project at the University of Melbourne [ICA]. This project was not based on XML, but also aimed at standardization of business terms for EDI in the form of repositories. The BSR was set up with multilinguality support. The semantics that it did provide (at least in the first prototype of which information is available) was limited; the main contribution was a structured way of describing EDI data elements. The idea behind BSI was that outgoing messages would be translated automatically to the standardized form at the sender’s site and translated back to the in-house file of the receiver at the receiver’s site. The translation in both cases is performed by an BSI server. Although we basically agree with this general idea, there are still many problems to be solved before this actually works. One is that a standard should be available that is sufficiently expressive - such as aimed at by XLBC and the multilingual thesaurus.

FIPA has taken the initiative of defining an Agent Communication Language [Fou]. The language is also based on Speech Act Theory and its semantics is specified in BDI logic. The FIPA language provides almost no support for conversation objects. The propositional content can be defined for different domains using a nested attribute/value scheme. The general message format is similar to XLBC; the main differences are that XLBC content is defined using predications and that the terminology is defined in a thesaurus (although FIPA also intends to set up directories where domain ontologies can be defined and published).

6 Conclusions and Future Work

The standardization process — defining a communication language and its semantics — is a process that is usually done by standardization committees, but if the users have to do it themselves, the question arises how it should be supported. We distinguish five aspects of this support:

Representation support: How to represent the syntax and semantics?
Accessibility support: How to store the definitions and make them available?
Methodological support: How to arrive at a definition of redefinition?
Process support: How to manage the standardization process?
Implementation support: How to implement the language in the context of existing legacy systems?

Drawing on FLBC, we introduced the XLBC language that defines the structure of messages. The meaning of the lexicals has to come from somewhere else. For this purpose, we have developed a multilingual thesaurus and an XLBC component library. By means of these techniques, the system is able to provide communication partners with representation and accessibility support, as mentioned above. Particularly, the specific (semantic) representation of business message components (in the shared repository) may incrementally evolve in our architecture. This would not be possible when relying only on prescribed XML DTD for message exchange. An important concern is the separation of the semantics in the repository and the concrete syntax in XML, as well as the separation of pragmatics (business rules on the application).

The other support aspects are not worked out in this paper, but we can make a few remarks. Process support is needed especially in the case that there are more than two stakeholders involved, for example, a business group or virtual community. In that case, the process should start by identifying all relevant stakeholders and ensure that everyone who wants to be involved has the possibility to do so. It is important that the process is legitimate so that the results are acceptable to all stakeholders. In [dM99], a method is described in which virtual professional communities can arrive at acceptable specifications. This method can be used also for a definition process.

Implementation support is especially important for the coupling of the standardized language with the legacy systems of the parties involved. Typically, the communication language is not identical to the language spoken by these legacy systems. A translation or mapping is needed to transform one representation into the other. This translation software is one of the major components of current EDI systems. [Has00] discusses the role of standards in the construction and mapping of global data models for cooperative information systems with different individual data models. The traditional bottom-up approach is to start with the data models to be integrated and then trying to define superclasses of which the original classes are specializations. The study shows that this can lead to very complex integrated models. A top-down approach starts with an available domain model, as the multilingual thesaurus may provide, and maps this to the situation at hand in the legacy systems. In the case of a message standard, a top-down approach could be followed if generic concepts, such as order, invoice but also product, buyer, seller, or transport medium are available. The top-down approach and the bottom-up approach can be combined in a so-called *yo-yo* approach. On the technical level, wrappers that provide unified interfaces are an established technique for accessing legacy systems [RS97].

Acknowledgments

The first version of XLBC and the associated Shared Repository have been developed and implemented in the context of the ESPRIT project MEMO (Mediating and Monitoring Electronic Commerce) with project code 26.895. The partners are: ABN-AMRO, Tilburg University (CentER Applied Research), RWTH Aachen, EURIDIS Rotterdam, IHK Aachen, EKD Spain, Sarenet Spain and Origin Spain. For more information, refer to
`http://www.abnamro.com/memo`.

References

[D+98] S. Decker et al. ONTOBROKER: Ontology based access to distributed and semi-structured information. In R. Meersman, Z. Tari, and S. Stevens, editors, *Proc. DS-8: Database Semantics - Semantic Issues in Multimedia Systems*. Kluwer Academic Publishers, 1998.

[Dik89] S.C. Dik. *The theory of Functional Grammar*. Mouton-De Gruyter, 1989.

[dM99] A. de Moor. *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*. PhD thesis, Tilburg University, 1999.

[Fou] Foundation for Intelligent Physical Agents. *FIPA*. (available from `http://www.fipa.org/`).

[G+99] M. Gisler et al. Requirements on secure electronic contracts. Technical Report MCM-institute-1999-01, Univ. St. Gallen, 1999.

[H+95] C.-F. Helgesson et al. Standards as institutions. problems with creating all-european standards for terminal equipment. In J. Groenewegen, C. Pitelis, and S.-E. Sjostrand, editors, *On Economic Institutions*, pages 162–184. Edward Edgar Publ., 1995.

[Has00] W. Hasselbring. The role of standards for interoperating information systems. In K. Jakobs, editor, *Information Technology Standards and Standardization: A Global Perspective*, pages 116–130. Idea Group Publishing, 2000.

[Hue98] C. Huemer. The growing need for meta messages in electronic data interchange. In *Proc. 6th ECIS*, Aix-en-Provence, France, 1998.

[ICA] ICARIS. *Business System Interoperation (BSI) – a Generic Solution*. (available from `http://www.cs.mu.oz.au/research/icaris/bsi.html`).

[ISO] ISO/UN-ECE. *The Basic Semantic Repository*. (available from http://www.cs.mu.oz.au/research/icaris/bsr.html).

[Kim98] S. Kimbrough. On ESTheta Theory and the Logic of X12 Date/Time Qualifiers. In *Proc. HICSS'98*, 1998.

[KL96] S.O. Kimbrough and R.M. Lee. On formal aspects of electronic commerce: examples of research issues and challenges. In *Proceedings HICSS'96*. IEEE Computer Society Press, 1996.

[KM97] S.O. Kimbrough and S.A. Moore. On automated message processing in electronic commerce and work support systems: speech act theory and expressive felicity. *ACM Transactions on Information Systems*, 15(4):321–367, 1997.

[Lee98] R. Lee. Towards open electronic contracting. *Electronic Markets*, 8(3), 1998.

[McG98] S. McGrath. *XML by example: building e-commerce applications*. Prentice Hall, Upper Saddle River, NJ, 1998.

[Mey97] B. Meyer. *Object-Oriented Software Construction*. Prentice Hall, Englewood Cliffs, NJ, second edition, 1997.

[MG98] B. Meltzer and R. Glushko. XML and electronic commerce enabling the network economy. *SIGMOD Record*, 27(4):21–24, 1998.

[Moo99] S.A. Moore. A foundation for flexible automated electronic communication. Working paper, University of Michigan Business School, 1999. (available from http://www-personal.umich.edu/˜samoore/research/flbc/).

[OIL] OIL (Ontology Inference Layer). See http://www.ontoknowledge.org/oil.

[RS97] M.T. Roth and P.M. Schwarz. Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *Proc. 23rd International Conference on Very Large Data Bases (VLDB'97), August 1997, Athens, Greece*, pages 266–275. Morgan Kaufmann, 1997.

[Wei89] H. Weigand. *Linguistically Motivated Principles of Knowledge Based Systems*. PhD thesis, Vrije Universiteit, Amsterdam, 1989.

[WVD95] H. Weigand, E. Verharen, and F. Dignum. Integrated semantics for information and communication systems. In R. Meersman and L. Mark, editors, *Proc. IFIP WG 2.6 Conf on Data Application Semantics*, pages 500–525. Chapman & Hall, 1995.

[WvH99] H. Weigand and W.-J. vd Heuvel. Meta-patterns for electronic commerce. *International Journal on Electronic Commerce*, 3(2):45–66, 1999.